

Московский государственный технический университет им. Н. Э. Баумана

Факультет Информатика и системы управления

Кафедра Теоретическая Информатика и Компьютерные Технологии

## ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

### «Формирование уникальных ядер в статистических тематических моделях»

Выполнил:

студент 4 курса группы ИУ9-81

*Ашуха Арсений Павлович*

Научный руководитель:

к.ф-м.н., профессор

*Лукашевич Наталья Валентиновна*

Москва, 2015

# Аннотация

*Вероятностное тематическое моделирование* позволяет выделить темы в коллекции документов и определить степень принадлежности каждого документа к каждой теме.

Стандартные подходы хорошо описывают коллекцию, но темы часто сильно похожи между собой и плохо интерпретируемы человеком, что вызывает проблемы во многих приложениях тематических моделей.

Стандартные подходы к тематическому моделированию, как правило, используют только частотные характеристики терминов, и совсем не учитывают лингвистическую информацию о строении текста. Добиться более высокой интерпретируемости и различности тем можно за счёт внедрения лингвистической информации в тематические модели.

В данной работе предложены подходы к учёту морфологии и словосочетаний для построения тематических моделей.

# Содержание

<b>Введение</b>	<b>9</b>
<b>1 Основы тематического моделирования</b>	<b>11</b>
1.1 Обозначения . . . . .	11
1.2 Гипотеза мешка слов . . . . .	11
1.3 Тема, Документ . . . . .	12
1.4 Гипотеза условной независимости . . . . .	12
1.5 Вероятностная модель коллекции документов . . . . .	13
1.6 Постановка задачи тематического моделирования . . . . .	13
1.7 Предобработка коллекции . . . . .	15
1.8 Метрики качества . . . . .	17
<b>2 Алгоритмы тематического моделирования</b>	<b>20</b>
2.1 PLSA . . . . .	20
2.2 Алгоритм Anchor Words . . . . .	23
2.3 Недостатки стандартных алгоритмов . . . . .	30
<b>3 Реализация</b>	<b>31</b>
3.1 Используемые инструменты . . . . .	31
3.2 Абстрактный класс <code>Collection</code> и его наследники . . . . .	32
3.3 Абстрактный класс <code>Transformer</code> и его наследники . . . . .	33
3.4 Стоп-слова . . . . .	35
3.5 Интерфейсы тематических моделей . . . . .	36
3.6 Недостатки реализации . . . . .	37
3.7 Процесс построения тематической модели . . . . .	38

<b>4</b>	<b>Учет лингвистических знаний</b>	
	<b>в тематических моделях</b>	<b>40</b>
4.1	Постановка задачи . . . . .	40
4.2	Комбинация Anchor Words и PLSA . . . . .	41
4.3	Поиск кандидатов в Anchor Words . . . . .	43
4.4	Учёт словосочетаний в Anchor Words . . . . .	45
4.5	Сравнение результатов . . . . .	49
	<b>Заключение</b>	<b>50</b>
	<b>Список литературы</b>	<b>51</b>

## Список Листингов

1	Абстрактный класс Collection . . . . .	32
2	Абстрактный класс Transformer . . . . .	34
3	Многопоточный аналог класса Transformer . . . . .	34
4	Последовательное применение цепочки преобразований . . . . .	34
5	Интерфейсы тематических моделей . . . . .	36
6	Интерфейс метрик качества тематических моделей . . . . .	37
7	Пример запуска эксперимента для построения модели . . . . .	39

## Список таблиц

1	Сравнительная таблица метрик качества различных алгоритмов	49
2	Пример тем, алгоритм PLSA . . . . .	53
3	Пример тем, алгоритм Anchor Words . . . . .	54
4	Пример тем, алгоритм Anchor Words + PLSA . . . . .	55
5	Пример тем, алгоритм Anchor Words + PLSA + Morph . . . . .	56
6	Пример тем, алгоритм Anchor Words + PLSA + Bigramm . . . . .	57

# Введение

Вероятностное тематическое моделирование — это современный статистический инструмент анализа коллекций документов различной природы, позволяющий выделить набор тем в коллекции документов, и описать каждый документ дискретным вероятностным распределением на множестве тем.

Понятие темы неоднозначно, обычно под темой понимают набор семантически связанных терминов, дискретное вероятностное распределение на множестве слов.

Тематическое моделирование активно применяется для различных прикладных задач:

- Поиск по запросу любой длины (абзац, глава, книга, ...) и природы (картинки, музыка, ...).
- Категоризация, классификация, аннотирование, суммаризация, сегментация текстовых документов.
- Анализ и агрегирование новостных потоков.
- Рубрикация документов, изображений, видео, музыки.
- Аннотация генома и другие задачи биоинформатики.
- Сжатие информации об объекте, выделение тематических групп пользовательских ресурсов.

Такое широкое применение возможно потому, что для использования алгоритмов тематического моделирования достаточно определить, что считать терминами, а что документами, при этом термины и документы не обязаны носить текстовый характер.

Стандартные алгоритмы при решении задачи тематического моделирования учитывают только частоты слов и не учитывают: синтаксическую структуру документа, порядок терминов в документе, морфологическую окраску терминов. Некоторые модели испытывают проблемы с выбором начального приближения или существенно уступают по качеству другим моделям.

Пренебрежение лингвистической информацией влечёт за собой не высокую интерпретируемость тем, низкую степень различия тем, а также ухудшение других метрик качества.

На практике низкие метрики качества затрудняют применение тематической модели: при низкой интерпретируемости некоторые темы могут показаться слабо связанным набором терминов или смесью неярко выраженных тем, если темы сильно похожи друг на друга, то набор тем в целом тяжело интерпретировать.

В данной работе предложена комбинация двух различных (непохожих друг на друга) алгоритмов тематического моделирования, добавлен учёт морфологической информации, а также словосочетаний.

Проведённые эксперименты подтвердили, что внедрение лингвистических знаний позволяет улучшить тематические модели, одновременно сразу по нескольким метрикам качества.

# 1 Основы тематического моделирования

В данном разделе вводятся общепринятая терминология и предположения *тематического моделирования*, раздел заканчивается постановкой задачи, *тематического моделирования*.

## 1.1 Обозначения

В работе используются следующие обозначения:

- $D, W, T$  – множества документов, слов и тем, иногда эти обозначения будут означать мощности множеств, что всегда понятно из контекста.
- $d, w, t$  – конкретный документ, слово или тема.
- $n_{i_1 \dots i_n}$  – число совместного появления  $i_1 \dots i_n$ .
- $p(a)$ ,  $\hat{p}(a)$  – вероятность и оценка вероятности события  $a$ .
- $G$  – матрица, то  $g_{ij}$  –  $i, j$  элемент матрицы  $G$ .

## 1.2 Гипотеза мешка слов

**Гипотеза мешка слов** заключается в предположении, что можно определить, какими темами образован документ, не учитывая порядок слов в документе [VoronSlides2015]. Тогда каждый документ можно представить в виде неупорядоченной последовательности слов, *мешка слов*. *Мешок слов* можно компактно хранить в виде набора пар  $d = \{(w, n_{dw}) : w \in W, n_{dw} \neq 0\}$ , где  $n_{wd}$  – количество вхождений слова  $w$  в документ  $d$ .



### 1.3 Тема, Документ

Темой, принято считать набор терминов, характеризующий предметную область, событие и так далее. При этом, понятие темы неоднозначно — если попросить двух разных людей составить набор терминов характеризующий тему *программирование*, то наборы терминов получатся несомненно разные. При этом, одно слово может принадлежать сразу нескольким темам, к примеру слово *ягуар* может принадлежать к теме: *животные*, *машины*, *напитки*.

Формализуя выше сказанное; **Тема** — дискретное вероятностное распределение на терминах, то есть тема  $t = \{P(w|t) : w \in W\}$  [VoronSlides2015]. При такой формализации каждое слово в каждой теме имеет некоторую вероятность, возможно нулевую. При этом удобно представлять вероятности слов в темах, как матрицу  $\Phi_{W \times T}$ .

**Документ** — дискретное вероятностное распределение на темах  $P(t|d)$ . При этом удобно представлять вероятности тем в документах, как матрицу  $\Theta_{T \times D}$  [VoronSlides2015].

### 1.4 Гипотеза условной независимости

**Гипотеза условной независимости** предполагает, что тема не зависит от документа, то есть представлена одним и тем же дискретным распределением в каждом документе содержащем данную тему [VoronSlides2015]. Формально говоря — вероятность встретить слово при условии темы не зависит от документа  $P(w|d, t) = P(w|t)$ .

## 1.5 Вероятностная модель коллекции документов

Под коллекцией документов подразумевается набор документов (текстовых, графических, ...). С вероятностной точки зрения, документы получены из дискретного вероятностного пространства троек [VoronSlides2015]

$$P\{D \times W \times T\} = \{(d, w, t) \sim p(d, w, t) : d \in D, w \in W, t \in T\} \quad (1)$$

где  $d, w$  — наблюдаемые координаты, а темы  $t$  — скрытые.

Тогда, с учётом *гипотезы условной независимости*, вероятность встретить слово  $w$  в документе  $d$ :

$$p(w|d) = \sum_{t \in T} P(w|d, t)P(t|d) = \sum_{t \in T} P(w|t)P(t|d) \quad (2)$$

**Вероятностный процесс порождения коллекции** можно представить следующим образом: когда автор создаёт новый документ, он выбирает вектор вероятностей тем в документе  $p(t|d)$ , а дальше для написания каждого слова с вероятностью  $p(t|d)$  выбирает тему  $t$ , после чего выбирает слово из темы  $t$  — распределения  $p(w|t)$ .

## 1.6 Постановка задачи тематического моделирования

Если вероятностная модель коллекции документов по известным темам и тематическим профилям документов порождает коллекцию, то построение тематической модели, задача обратная — по известной коллекции восстановить темы, породившие коллекцию и тематические профили документов её образующих.

Обозначим,  $\Phi_{W \times T} = p(w|t)_{W \times T}$ ,  $\Theta_{T \times D} = p(t|d)_{T \times D}$ , причем  $F_{W \times D} \approx \Phi\Theta \approx p(w|d)_{W \times D}$ , тогда задача построения матриц  $\Phi$  и  $\Theta$  сводится к следующей задаче матричного разложения с ограничениями [VoronSlides2015]:

Известна матрица  $F$ , найти матрицы  $\Phi$  и  $\Theta$ :

$$F \approx \Phi\Theta \quad (3)$$

при вероятностных ограничениях неотрицательности и нормировки:

$$\phi_{wt} \geq 0; \quad \sum_{w \in W} \phi_{wt} = 1; \quad \theta_{td} \geq 0; \quad \sum_{t \in T} \theta_{td} = 1. \quad (4)$$

Заметим, что задача разложения  $F \approx \Phi\Theta$  некорректно поставлена [VoronSlides2015], поскольку  $F \approx \Phi\Theta \approx (\Phi S)(S^{-1}\Theta) = \Phi'\Theta'$  при  $S_{T \times T}$  таких, что для матриц  $\Phi', \Theta'$  выполняются условия 4.

Многоекстремальность и многокритериальность задачи, делает её достаточно сложной с точки зрения методов оптимизации.

Наглядно, тематическую модель можно представить следующим образом (Рис. 1) — документ, это последовательность тематически окрашенных терминов, который можно представить в виде дискретного распределения на темах, принадлежности к теме выделена цветом. В свою очередь — тема набор терминов, упорядоченный в порядке возрастания вероятности термина в теме.

Заметим, что с точки зрения постановки задачи, термин может иметь как нулевую вероятность во всех темах — не являться тематически окрашенным, так и высокую вероятность во всех темах — являться термином общей лексики.

Матричную форму записи тематической модели иллюстрирует Рис. 2. Предполагается, что матрицы  $\Phi, \Theta, F$  стохастические, то есть выполнены условия 4.

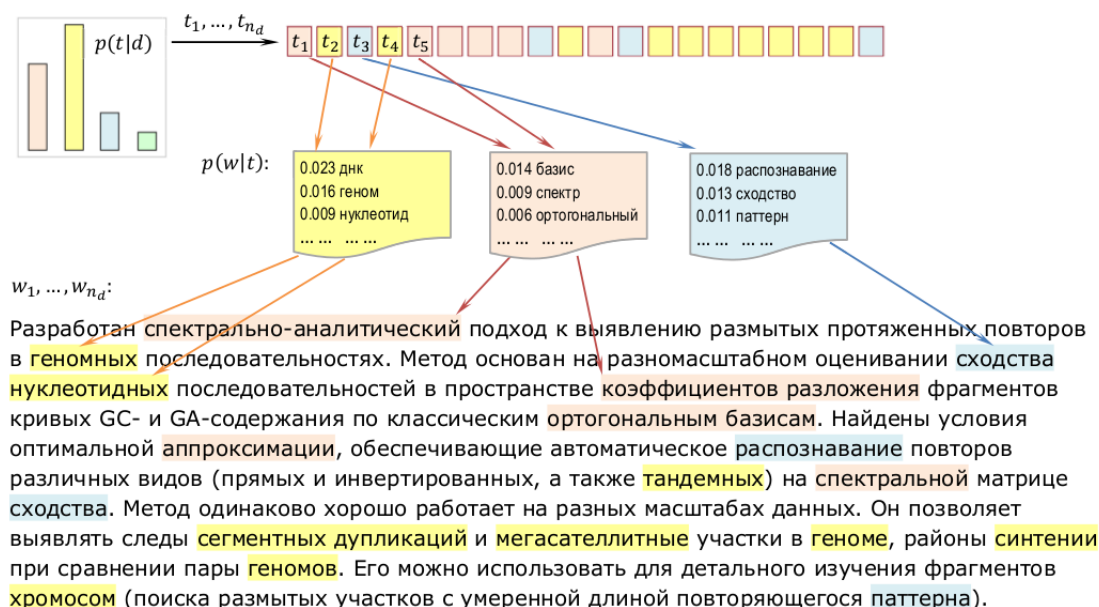


Рис. 1: Пример выделения тем в тексте ([ru.wikipedia.org/wiki/Тематическое\\_моделирование](http://ru.wikipedia.org/wiki/Тематическое_моделирование)).

## 1.7 Предобработка коллекции

Предобработка коллекции предполагает удаление «лишней» информации, которая, как предполагает исследователь, негативно сказывается на качестве модели.

Избыточное количество параметров модели, как правило, влечёт за собой переобучение и медленную настройку модели. Именно поэтому большинство преобразований направлены на уменьшение параметров тематической модели, обычно за счёт уменьшения размера словаря. Некоторые примеры популярных преобразований:

- На основе предположения, что редкие термины не могут быть яркими представителями какой-либо темы, производят удаление редких терми-

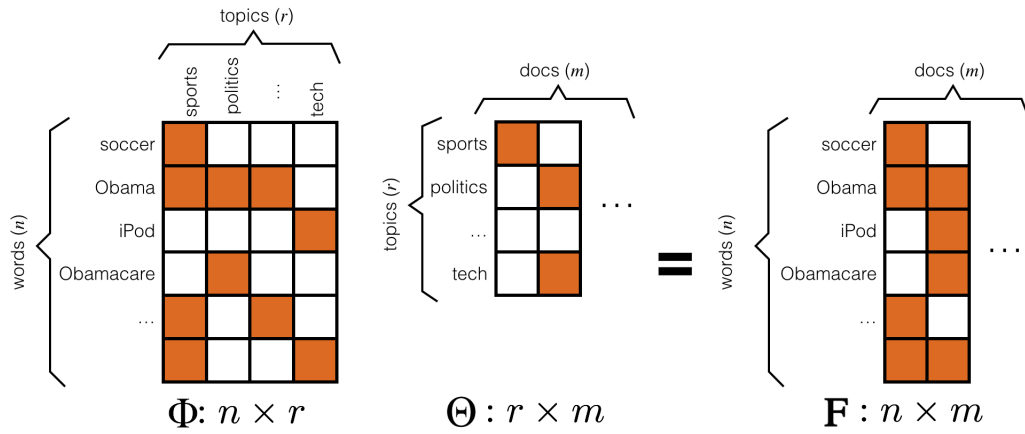


Рис. 2: Пример записи тематической модели в матричной форме [Voss2014]

нов, что существенно сокращает размер коллекции, потому что редкие термины составляют 85 – 90% словаря коллекции. Уменьшение словаря положительно сказывается на метриках качества модели. В то же время такое преобразование усугубляет проблему полноты, повышает количество незнакомых модели слов.

- На основе предположения, что разные формы одного и того же термина не влияют на тематическую окраску, термины приводят к нормальной форме, что так же уменьшает словарь модели. Процесс нормализации позволяет убрать из исходного текста грамматическую информацию (падежи, числа, Введение глагольные виды и времена, залоги причастий, род и так далее), оставляя смысловую составляющую.
- На основе предположения, что некоторые части речи (междометия, предлоги, союзы, частицы, местоимения) не несут в себе тематическую окраску, с целью уменьшения размера словаря и коллекции, слова таких частей речи можно удалить.

- На основе предположения, что термины общей лексики (стоп-слова) не несут в себе тематической окраски, их также можно удалить.
- Иногда предполагается, что из коллекции выделены словосочетания и добавлены в коллекцию, как уникальные элементы словаря. Обычно значение такого термина отлично от значений слов его образующих. К примеру *Машина опорных векторов*, имеет посредственное отношение к *машинам*, *опорам* или *векторам*.
- Пренебрегают знаками пунктуации и непечатными символами, приводят все термины к одному регистру.

## 1.8 Метрики качества

Задача тематического моделирования некорректно поставлена, в отличие от задач классификации или регрессии здесь нет чёткого понятия «ошибки» или «потери». Стандартные критерии качества кластеризации плохо подходят для оценивания «мягкой» совместной кластеризации документов и терминов.

Поэтому нет возможности предложить одну метрику, которая сможет дать полную информацию о качестве тематической модели. Из всех предложенных сообществом метрик, как правило, стандартом являются: перплексия и когерентность [VoronSlides2015].

### Перплексия

Наиболее распространённым критерием является перплексия (perplexity), используемая для оценивания моделей языка в компьютерной лингвистике. Это мера несоответствия или «удивлённости» модели  $p(w|d)$  терминам  $w$ , наблю-

даемым в документах  $d$  коллекции  $D$ , определяемая через логарифм правдоподобия:

$$LogLikelyHood(D, \Phi, \Theta) = \sum_{d \in D} \sum_{w \in d} n_{dw} \ln \sum_{t \in T} \phi_{wt} \theta_{td} \quad (5)$$

$$Perplexity(D, \Phi, \Theta) = \exp \left( -\frac{1}{len(D)} LogLikelyHood(D, \Phi, \Theta) \right) \quad (6)$$

Чем меньше эта величина, тем лучше модель  $p$  предсказывает появление терминов  $w$  в документах  $d$  коллекции  $D$ .

Перплексия чувствительна к размеру словаря, что делает её неприменимой для сравнения тематических моделей с различным словарём. Эта проблема решена в [Nyzhybytskyu2014].

Поскольку перплексия определена через правдоподобие, то величина перплексии сама по себе не несёт информации о качестве модели, но сравнивать перплексии разных моделей — корректно.

## Когерентность

Когерентность (coherence) – мера интерпретируемости тем, предложенная в [Newman2009]. Когерентность показала высокую корреляцию с оценками экспертов.

$$Coherence(D, Topic) = median \left( \log \frac{P(w_1, w_2)}{P(w_1)P(w_2)} : w_i, w_j \in Topic_{1,...,10} \right) \quad (7)$$

На самом деле, с точки зрения теории информации когерентность является поточечной взаимной информацией. Чем эта метрика больше, тем тема более интерпретируема. В то же время высокая когерентность не гарантирует, что тема будет хорошей с точки зрения семантической связности, это

значит, что когерентность необходимо использовать вместе с перплексией. Общая когерентность модели определяется как медиана когерентностей тем.

## Мера уникальности ядер

Исходя из предположения, что темы выделенные человеком в коллекции как правило будут иметь уникальное ядро – набор терминов с высокой вероятностью в данной теме. Мера уникальности ядер показывает насколько темы не похожи между собой.

$$KernelsUniq = \frac{|uniq(kernels)|}{|kernels|} \quad (8)$$

Чем уникальность ядер ближе к единице, тем легче различить темы, чем уникальность ближе к нулю, тем сложнее разделить темы. В данной работе ядро темы — набор десяти наиболее вероятных терминов в теме.



## 2 Алгоритмы тематического моделирования

В данном разделе, приведён краткий обзор основных методов и подходов к решению задачи *тематического моделирования*. Основным алгоритмом является PLSA (ARTM и PLSA-SIM являются его модификациями), существенно отличным от PLSA является алгоритм ANCHOR WORDS.

### 2.1 PLSA

**Probabilistic Latent Semantic Analysis** (Вероятностный латентный семантический анализ, PLSA) был предложен Томасом Хофманном в [Hofmann1999].

Для построения модели, предполагается оптимизировать логарифм правдоподобия, одновременно с этим оптимизируя перплексию,

$$LogLikelyHood(D, \Phi, \Theta) = \sum_{d \in D} \sum_{w \in d} n_{dw} \ln \sum_{t \in T} \phi_{wt} \theta_{td} \rightarrow \max_{\Phi, \Theta}, \quad (9)$$

при ограничениях нормировки и неотрицательности 4.

Для решения поставленной оптимизационной задачи предлагается использовать ЕМ-алгоритм, используемый в математической статистике для нахождения оценок максимального правдоподобия параметров вероятностных моделей, в случае, когда модель зависит от некоторых скрытых переменных, каждая итерация которого состоит из двух шагов – Expectation, Maximization.

На Е-шаге по текущим значениям параметров  $\phi_{wt}$ ,  $\theta_{td}$  с помощью формулы Байеса вычисляются условные вероятности  $p(t|d, w)$  всех тем  $t \in T$  для каждого термина  $w \in d$  в каждом документе  $d$ :

$$H_{dwt} = p(t|d, w) = \frac{p(w|t)p(t|d)}{p(w|d)} = \frac{\phi_{wt}\theta_{td}}{\sum_{t' \in T} \phi_{wt'}\theta_{t'd}} \quad (10)$$

На М-шаге, наоборот, при фиксированных вероятностях тем  $H_{dwt}$  вычисляются оценки максимального правдоподобия для параметров  $\phi_{wt}$ ,  $\theta_{td}$ :

$$\phi_{wt} = \frac{n_{wt}}{n_t}, \quad n_{wt} = \sum_{d \in D} n_{dw} H_{dwt}, \quad n_t = \sum_{w \in W} n_{wt} \quad (11)$$

$$\theta_{td} = \frac{n_{dt}}{n_d}, \quad n_{dt} = \sum_{w \in d} n_{dw} H_{dwt}, \quad n_d = \sum_{t \in T} n_{td} \quad (12)$$

Перед началом работы алгоритма необходимо задать начальное приближение  $\Phi$  и  $\Theta$ , что можно сделать нормированными случайными векторами. Но на самом деле выбор хорошего начального приближения является отдельной сложной задачей.

Более того, PLSA достаточно чувствителен к выбору начального приближения — изменяя только начальное приближение можно получить существенно различные темы и качество модели.

---

**Алгоритм 1** ЕМ-алгоритм для тематической модели PLSA.

---

**Вход:** коллекция  $D$ , число тем  $|T|$ , начальные приближения  $\Phi$  и  $\Theta$ ;

**Выход:** распределения  $\Phi$  и  $\Theta$ ;

- 1: **повторять:**
  - 2:   обнулить  $n_{wt}$ ,  $n_{dt}$ ,  $n_t$  для всех  $d \in D$ ,  $w \in W$ ,  $t \in T$
  - 3:   **для всех**  $d \in D$ ,  $w \in d$ :
  - 4:      $Z = \sum_{t \in T} \phi_{wt}\theta_{td}$ ;
  - 5:     **для всех**  $t \in T$  таких, что  $\phi_{wt}\theta_{td} > 0$  :
  - 6:       увеличить  $n_{wt}$ ,  $n_{dt}$ ,  $n_t$  на  $\delta = n_{dw}\phi_{wt}\theta_{td}/Z$ ;
  - 7:    $\phi_{wt} = n_{wt}/n_t$  для всех  $w \in W$ ,  $t \in T$ ;
  - 8:    $\theta_{td} = n_{td}/n_d$  для всех  $d \in D$ ,  $t \in T$ ;
  - 9: **пока**  $\Phi$  и  $\Theta$  не стабилизируются
-

Сложность по времени EM-алгоритма 5 для решения задачи PLSA,  $O(Iter \cdot T \cdot [len(D) + W + D])$ , сложность по памяти зависит от эффективности хранения матриц  $\Phi$ ,  $\Theta$ , и в самом худшем случае составляет  $O(T(W + D))$ .

### **Аддитивная регуляризация тематических моделей**

Задачи матричного разложения, как правило нуждаются в регуляризации, чтобы избежать переобучения, в свою очередь регуляризация может как быть совершенно искусственной, к примеру как распределение Дирихле в модели «Латентное размещение Дирихле» (latent Dirichlet allocation, LDA)[Blei2003], так и нести в себе лингвистический смысл как «Аддитивная регуляризация тематических моделей, ARTM» [ARTM].

В работах [ARTM, Voron2013, VoronSlides2015] предложена теория ARTM. В приведённых работах предложены регуляризаторы, направленные на выполнение некоторых лингвистических требований к строению тематической модели: разреживание, сглаживание, декорреляция тем и другие. Причём записывать такие регуляризаторы можно не прибегая к использованию теории *Байесовского вывода*, что во многом упрощает и ускоряет разработку новых регуляризаторов.

С помощью аддитивной регуляризации (комбинации нескольких регуляризаторов), были устранены многие недостатки PLSA.

### **Учёт сходства между униграммами и биграммами**

Оригинальные тематические модели (PLSA и LDA) используют модель *мешка слов*, предполагающую независимость слов друг от друга. Однако в документах есть много слов, связанных между собой по смыслу, в частности

однокоренных, например: банк – банковский – банкир, кредит – кредитный – кредитовать – кредитование и др.

В работе [Nokel2014] предложена техника (модификация PLSA) учёта в тематических моделях похожих униграм и биграмм, с помощью которой было получено значительное улучшение метрик качества модели.

## 2.2 Алгоритм Anchor Words

Известно, что для невыпуклых функций задача поиска глобального экстремума является NP-полной [Kreinovich2005], поэтому возникает вопрос, можно ли решить задачу 3 не прибегая к принципу максимума правдоподобия.

В работе [Arora2012] был предложен алгоритм Anchor Words при наличии некоторых условий решающий задачу 3 за полиномиальное время, но фактическая сложность была очень высокой, настолько, что предложенный алгоритм был неприменим на практике.

Алгоритм, приведённый в [Arora2012], решает многочисленные линейные задачи, а также требует наложения определённых ограничений на искомые темы, использует построение обратных матриц, что приводит к неустойчивости алгоритма и возможности появления отрицательных вероятностей в стохастических матрицах.

После чего в статье [Arora2012b] был предложен модифицированный алгоритм Anchor Words, дана простая интерпретация «якорного разложения» и так же были предложены некоторые не столь значительные оптимизации алгоритма Anchor Words.

В итоге алгоритм оказался в значительно быстрее, чем PLSA-подобные алгоритмы и не унаследовал проблему с выбором начального приближения.

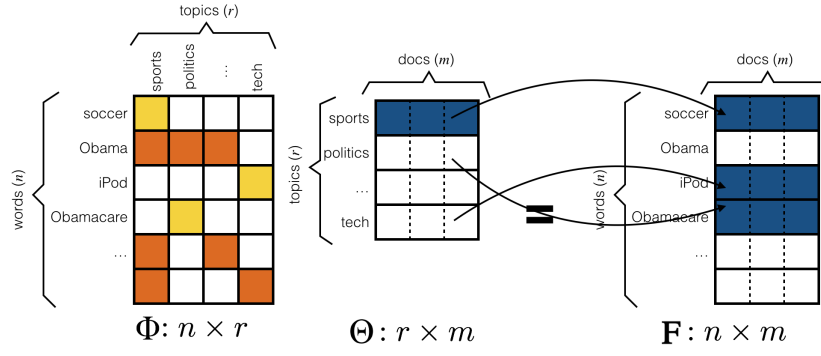


Рис. 3: p-разделимая матрица, [Voss2014]

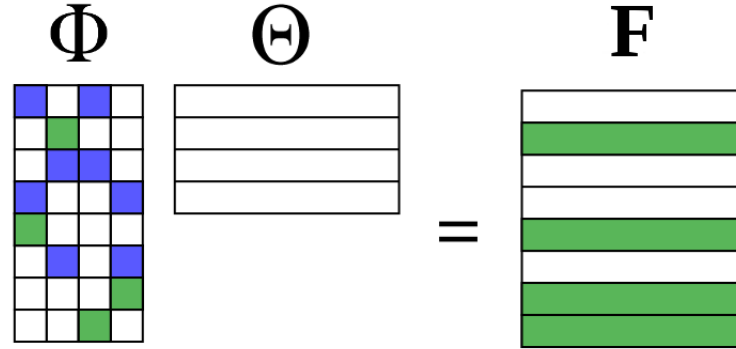


Рис. 4: p-разделимая матрица, [AnkurMoitraSlides2012].

## Основная идея алгоритма Anchor Words

Если в матрице  $\Phi$  для каждой темы существует слово, которое только в этой теме имеет ненулевую вероятность большую чем  $p$ , то будем называть такую матрицу p-разделимой (Рис. 3).

Пусть матрица  $\Phi$  — p-разделима, тогда в матрице  $F$  существуют строки, которые являются масштабированными копиями строк из  $\Theta$  (Рис. 4). Будем называть такие строки «якорными». Причём, из свойств матричного перемножения все остальные строки матрицы  $F$  являются линейной комбинацией якорных строк [Aroga2012].

Каждая «якорная» строка соответствует некоторой теме, и набор «якорных» строк является базисом, откуда логично получить разложения строковых векторов остальных слов по якорным (базисным) строкам (Рис. 5). Якорные

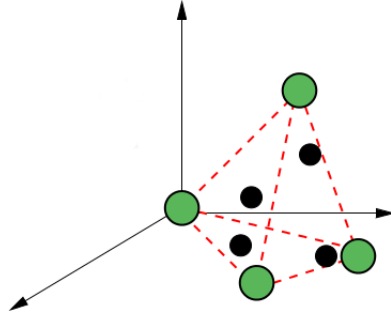


Рис. 5: Разложения по якорным словам, [AnkurMoitraSlides2012].

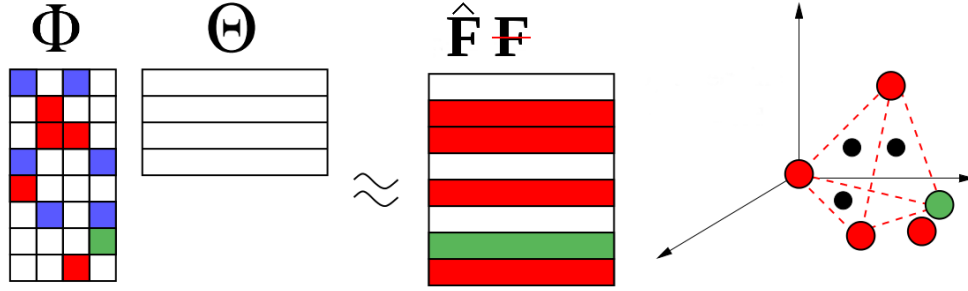


Рис. 6: Неустойчивость матрицы  $F$ , [AnkurMoitraSlides2012].

слова можно получить как выпуклую оболочку строк матрицы  $F$ .

Но есть проблема, вместо матрицы  $F$  мы имеем приближение  $\hat{F}$  которая на практике не устойчива и может плохо аппроксимировать  $F$  (Рис. 6) [Arora2012].

В [Arora2012] было предложено, использовать вместо матрицы  $\hat{F}$  матрицу  $FF^t$  (Рис. 7),

$$\hat{F}\hat{F}^t \rightarrow FF^t. \quad (13)$$

К счастью, то же свойство линейной комбинации строк сохраняется и для  $FF^t$ . Поэтому возможно восстановить якорные слова из  $FF^t$ . Так же заметим, что  $\Theta\Theta^t \rightarrow R$ ,  $R_{T \times T}$  можно рассматривать как ковариационную матрицу тем.

---

**Алгоритм 2** Высокоуровневый алгоритм Anchor Words.

---

**Вход:** коллекция  $D$ , число тем  $|T|$

**Выход:** матрица  $\Phi$ ;

- 1:  $Q = \text{Word Co-occurences}(D)$
  - 2:  $\hat{Q} = \text{Rows normalized } Q$
  - 3:  $S = \text{FindAnchorWords}(\hat{Q}, |T|)$
  - 4:  $\Phi = \text{RecoverWordTopic}(\hat{Q}, S)$
  - 5: **Вернуть**  $\Phi$
- 

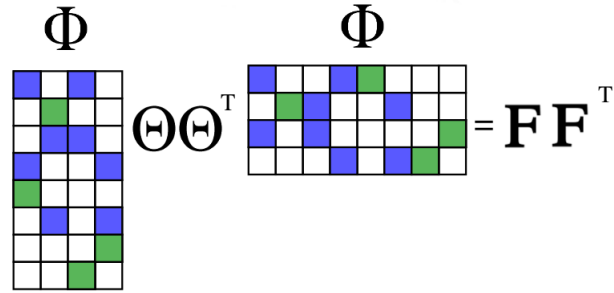


Рис. 7: Матрица  $FF^t$  до переупорядочивания строк (столбцов) в  $\Phi, \Theta$ , [AnkurMoitraSlides2012].

### Алгебраический подход к использованию $r$ -разделимости

В этом разделе необходимо понять, как можно использовать  $r$ -разделимость и якорные слова для поиска матриц  $\Phi$  и  $\Theta$ .

В [Aroga2012] приведён алгоритм решающий эту задачу. Переупорядочим строки и столбцы матрицы  $F$ , так, чтобы якорные строки и столбцы стояли в верхнем левом углу Рис. 8.

По правилам блочного перемножения матриц

$$Q = F\hat{F}^t = \Phi R \Phi^t = \begin{pmatrix} D \\ U \end{pmatrix} R \begin{pmatrix} D & U^t \end{pmatrix} = \begin{pmatrix} DRD & DRU^t \\ URD & URU^t \end{pmatrix}, \quad (14)$$

тогда требуется решить следующую задачу – по блокам, образующим матрицу  $Q$ , найти матрицы  $D, U$ .

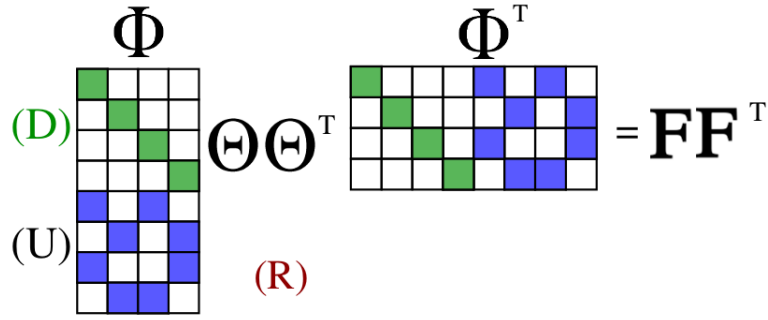


Рис. 8: Матрица  $\mathbf{F}\mathbf{F}^t$  после переупорядочивания строк (столбцов) в  $\Phi, \Theta$ , [AnkurMoitraSlides2012], [Arora2012].

---

**Алгоритм 3** Алгебраический алгоритм RecoverWordTopic.

---

**Вход:** матрица  $Q$ , множество якорных слов  $S$

**Выход:** матрица  $\Phi$ ;

- 1: Переставим строки и столбцы в матрице  $Q$ , так чтобы якорные строки образовывали диагональную подматрицу.
  - 2:  $p_S = Q_S \vec{1}$
  - 3:  $z = \text{solve}_z(Q_{S,S} \cdot z = p_S)$
  - 4:  $\Phi^t = (Q_{S,S} \cdot \text{Diag}(\vec{z}))^{-1} \cdot Q_S^t$
  - 5: **Вернуть**  $\Phi$
- 

Как уже было сказано ранее, этот алгоритм неприменим на практике.

## Вероятностный подход к использованию p-разделимости

В дополнение к недостаткам, описанным выше, хочется заметить, что алгебраический подход использует очень маленькую часть матрицы  $F$ , полагаясь только на совместную встречаемость слов и якорных слов, а эта оценка может быть неточной, если какое либо слово встречается нечасто. Заметим, что матрица  $Q_{ij} = p(w_i, w_j)$  нормализованная по строкам может быть рассмотрена как матрица условных вероятностей  $\bar{Q}_{ij} = p(w_i | w_j)$  [Arora2012b].

Обозначим множество индексов якорных слов как  $S = \{s_1, \dots, s_t\}$ . Якорные слова отличаются тем, что соответствующий набор строк из  $\bar{Q}_S$  образуют выпуклую оболочку строк матрицы  $\bar{Q}$ .



Докажем это. заметим, что для якорных слов

$$\bar{Q}_{S_t,j} = \sum_{t'} p(t'|s_t) \cdot p(w_j|t') \quad (15)$$

$$= p(w_j|t) \quad (16)$$

где 15 верно в силу свойств матричного умножения, а 16 потому, что  $p(t|s_t) = 1$ . Для всех остальных слов

$$\bar{Q}_{i,j} = \sum_{t'} p(t'|w_i) \cdot p(w_j|t') \quad (17)$$

Обозначим вероятность  $p(t|w_i)$  как  $C_{i,t}$ , тогда

$$\bar{Q}_{i,j} = \sum_{t'} C_{i,t'} \bar{Q}_{s_{t'},j}, \quad \sum_{t'} C_{i,t'} = 1 \quad (18)$$

таким образом строки из  $\bar{Q}$  являются линейной комбинацией якорных строк.

После чего пользуясь теоремой Байеса

$$p(w_i|t) = \frac{p(t|w_i)p(w_i)}{\sum_{i'} p(t|w_{i'})p(w_{i'})} \quad (19)$$

можем восстановить матрицу  $\Phi$ .

Если для решения оптимизационной задачи 20 использовать метрику  $L_2$

$$L_2(\bar{Q}_i, C_i^t \bar{Q}_S) = \|\bar{Q}_i - C_i^t \bar{Q}_S\|^2 = \|\bar{Q}_i\|^2 - 2C_i(\bar{Q}_S \cdot \bar{Q}_i^t) + C_i^t(\bar{Q}_S \cdot \bar{Q}_S^t)C_i, \quad (21)$$

то вычисление метрики на каждом шаге можно организовать независимо от размера словаря, поскольку  $\bar{Q}_S \cdot \bar{Q}_S^t$  может быть посчитана только один раз и

---

**Алгоритм 4** Алгоритм RecoverWordTopic.

---

**Вход:** Матрица  $Q$ , множество якорных слов  $S$ ;

**Выход:** матрица  $\Phi$ ;

1:  $\bar{Q} = \text{row\_norm}(Q)$

2:  $p_w = Q\vec{1}$  ▷ Сохраним нормировочные константы

3: **для всех**  $w \in W$ :

4:     Решить задачу:

$$C_i = \underset{\vec{C}_i}{\operatorname{argmin}} \operatorname{Measure}(\bar{Q}_i, \sum_{t \in T} C_{i,k} \cdot \bar{Q}_{s_k}) \quad (20)$$

5:     При ограничениях:  $\sum_{t \in T} C_{i,k} = 1, C_{ik} \geq 0$

6:  $A' = \operatorname{diag}(\vec{p}_w)C$

7:  $A = \operatorname{col\_norm}(A')$

8: **Вернуть**  $A$

---

использоваться для всех слов, а  $\bar{Q}_S \cdot \bar{Q}_i^t$  может так же быть вычислена только один раз, перед запуском алгоритма для слова  $w_i$ .

### Комбинаторный алгоритм для поиска якорных слов

Комбинаторный алгоритм, представленный в [Aroga2012b] приведён вместе с достаточно сложными математическими оценками, в работе будет приведён только сам алгоритм и приведены некоторые неформальные пояснения, оценки корректности и времени работы будут опущены.

При описании алгоритма  $\operatorname{span}(S)$  будет означать подпространство натянутое на вектора  $S$ . Расстояние от точки  $x$  до  $\operatorname{span}(S)$  вычисляется как норма проекции  $x$  на ортогональное дополнение  $\operatorname{span}(S)$ .

Поскольку число векторов в  $V$  равно его размерности необходимо снизить размерность пространства, уменьшение размерности позволяет избежать переобучения, уменьшить вычислительную сложность алгоритма.

---

**Алгоритм 5** Комбинаторный алгоритм FastAnchorWords.

---

**Вход:** Точки  $V = v_1, \dots, v_n$ , количество векторов для выпуклой оболочки –  $K$ ;

**Выход:**  $\{v'_1, \dots, v'_k\}$  – точки образующие наиболее выпуклую оболочку;

- 1: Спроецировать  $v_i$  на случайно выбранное пространство  $V$ ,  $\dim V = 4 \log V / \epsilon^2$
  - 2:  $S = \{s_0\}$ ,  $s_0$  – наиболее удалённая точка от начала координат.
  - 3: **для всех**  $i = 1$  to  $K$ :
  - 4:     Обозначим  $s_i$  точку из  $V$  наиболее удалённую от  $\text{span}(S)$ .
  - 5:      $S = S \cup \{s_i\}$
  - 6: **для всех**  $i = 1$  to  $K$ :
  - 7:     Обозначим  $s'_i$  точку из  $V$  наиболее удалённую от  $\text{span}(S \setminus \{s_i\})$ .
  - 8:     Заменить  $s_i$  на  $s'_i$
- Вернуть**  $S$
- 

## 2.3 Недостатки стандартных алгоритмов

1. Бедная модель документа – модель *мешка слов*, не учитывает совместную встречаемость слов, структуру предложений документа, в следствие чего теряет очень много информации.
2. Не используются даже простые предположения о структуре тематически окрашенных терминов – высокую вероятность в теме может получить совершенно любой термин, так как при построении модели алгоритм руководствуется только частотами слов, но никак не учитывает их лексическую или грамматическую структуру.
3. Все модели, основанные на модели PLSA, неустойчивы к выбору начального приближения.
4. Модель Anchor Words, как правило, обладает более высокой, перплексией чем PLSA.

## 3 Реализация

В текущем разделе будут описаны инструменты и некоторые детали реализации системы для проведения экспериментов.

### 3.1 Используемые инструменты

Для реализации тематических моделей был использован язык `python`, в настоящее время `python` (<https://www.python.org/download/releases/2.7/>) является наиболее используемым языком общего назначения для проведения научных исследований.

За исключением стандартных пакетов, были использованы следующие пакеты (<https://pypi.python.org/pypi/pip>):

1. `ipython notebook` — современная интерактивная среда для программирования.
2. `nltk` — пакет для обработки естественного языка, содержит различные синтаксические и семантические анализаторы.
3. `scipy` — реализация высокоуровневых математических операций (обработка разреженных матриц, методы оптимизации, ...)
4. `numpy` — реализация низкоуровневых математических операций, быстрая работа с матрицами и векторами.
5. `py morphology` — морфологический и семантический анализ для текстов на русском языке.

Реализация экспериментов и алгоритмов доступна в открытом репозитории по адресу [goo.gl/f8ZdcW](https://goo.gl/f8ZdcW).

## 3.2 Абстрактный класс Collection и его наследники

Для задачи обработки текстов на естественном языке, не существует единого формата хранения текстовых коллекции. Обычно каждый формат хранения предпочтительный для конкретной задачи имеет свои плюсы и минусы. Поэтому, было принято решение на этапе загрузки, хранить коллекцию наиболее очевидным образом: в каждой коллекции представленной классом `Collection` (Листинг. 1), документы разделяются на два подмножества для *обучения* и *контроля*. Каждый подмножество, это список документов, а каждый документ список терминов.

```
1 class Collection():
2     def __init__(self, path, lang):
3         self.documents_train = list()
4         self.documents_test  = list()
5
6     def fill(self):
7         raise NotImplementedError
8
9     def save(self):
10        ...
11
12    def tobow(self):
13        ...
```

Листинг 1: Абстрактный класс Collection

Ввиду огромного множества форматов, у класса `Collection` метод `fill` является абстрактным, и реализуется отдельно для каждого формата коллекции в наследниках класса `Collection`.

Были реализованны следующие наследники класса `Collection`:

1. `FullTextCollection` — каждый документ представляется в виде упорядоченного набора терминов, учитывается порядок терминов.

2. `NltkCollection` — достаточно сложный формат хранения, предполагает нетривиальную лемматизацию, но в таком формате доступны достаточно много англоязычных коллекций.
3. `BagOfWordsCollection` — достаточно компактный формат коллекции, не учитывающий порядок терминов в документах, который не подходит польза для оценки частоты встречаемости терминов в документах.
4. `CompactTextCollection` — формат компактного хранения полнотекстовой коллекции, предполагает, что в коллекции выделен словарь, который сопоставляет каждому термину уникальный идентификатор, а документ, это последовательность заданных идентификаторов. Такую коллекцию можно представить достаточно компактно при условии небольшого размера словаря.

После считывания, любая коллекция приводится к формату компактного представления коллекции — `CompactTextCollection`. Также коллекцию можно представить в формате мешка слов, который является основным для построения тематических моделей, но не подходит для подсчёта некоторой дополнительной статистики.

### 3.3 Абстрактный класс `Transformer` и его наследники

Для обработки коллекции как правило необходимо произвести некоторый предподсчёт на основе которого необходимо совершить некоторые изменения над коллекциями. Поэтому был предложен абстрактный класс `Transformer` (Листинг. 2) с двумя методами `train` и `apply`.

```

1 class Transformer():
2     def train(self, collection):
3         raise NotImplementedError
4
5     def apply(self, collection):
6         raise NotImplementedError

```

Листинг 2: Абстрактный класс Transformer

Ввиду того, что некоторые преобразования независимы по данным, для каждого документа, был предложен многопоточный аналог класса Transformer — MultiThreadTransformer (Листинг. 3).

```

1 class MultiThreadTransformer():
2     def __init__(self, core=1):
3         self.core = core
4         self.map = None
5
6     def apply(self, collection):
7         pool = Pool(self.core)
8         collection.documents = pool.map(
9             self.map, collection.documents)

```

Листинг 3: Многопоточный аналог класса Transformer

Для последовательного применения преобразований был создан класс TransformerChainApply 4, который последовательно обучает и применяет список преобразований transformers.

```

1 class TransformerChainApply():
2     def __init__(self, transformers):
3         self.transformers = transformers
4
5     def apply(self, collection):
6         ...

```

Листинг 4: Последовательное применение цепочки преобразований

Были реализованны следующие наследники класса `Transformer`:

1. `PunctuationRemoverTransform` — удаление пунктуации и непечатных символов.
2. `LoweCaseTransform` — приведение всех символов к нижнему регистру.
3. `WordNormalizerTransform` — приведение слов к нормальной форме.
4. `StopWordsRemoverTransform` — удаление стоп-слов русского и английского языка.
5. `BigramExtractorDocumentsTransform` — извлечение словосочетаний.
6. `TrashFilterTransform` — удаление редких слов.
7. `HtmlClearTransform` — очистка html-разметки.
8. `WikiClearTransform` — очистка от wiki-разметки.

Также в процессе написания программы были реализованы другие наследники класса `Transform`, но не использовались в финальной версии программы и были удалены.

### 3.4 Стоп-слова

Для тематического моделирования, стоп-словами являются слова не имеющие яркой тематической окраски, в виду чего были удалены:

Были удалены:

1. **Слова общей лексики:** есть, еще, ж, же, за, зачем, здесь, ...
2. **Короткие слова:** а, и, ах, как, он, она, нас, ...



3. **Редкие слова:** венчурный, гривенник, диспропорция, ...
4. **Все части речи кроме:** существительных, прилагательных, наречий и глаголов.
5. **Слова, которые встречались только в одном тексте**

### 3.5 Интерфейсы тематических моделей

В рамках дипломной работы были реализованы тематические модели PLSA и Anchor Words. Модели реализованны в виде обычных функций которые принимают на вход коллекцию текстовых документов и параметры алгоритма а возвращают тематическую модель. Интерфейсы моделей приведены в Листинге 5.

```
1 def plsa_model(collection, wrd_count, num_topics=100, num_iter=10,  
2               metrics=None, verbose=False, F=None):  
3     ...  
4  
5 def anchor_model(collection, wrd_count, num_topics=100,  
6                  metrics=None, k=400, verbose=False,  
7                  noun=False, bi=False):  
8     ...
```

Листинг 5: Интерфейсы тематических моделей

Описание параметров тематических моделей:

1. `collection` — коллекция текстовых документов экземпляр класса `Collection`.
2. `wrd_count` — размер словаря.
3. `num_topics` — число тем.

4. `num_iter` — число итераций.
5. `metrics` — список метрик, функций с интерфейсом 6.
6. `k` — параметр алгоритма Anchor Words, накладывающий частотное ограничение на якорные слова.
7. `verbose` — регулирует подробность вывода состояния модели в ходе выполнения алгоритма.
8. `noun` — ключик, якорные слова могут быть только существительными
9. `bi` — использовать биграммы для поиска тематической модели.
10. `F` — начальное приближение матрицы  $F$  в алгоритме PLSA.

```
1 def measure(word_topic_matrix, train, test):  
2     ...
```

Листинг 6: Интерфейс метрик качества тематических моделей

Были реализованы следующие метрики:

1. Когерентность
2. Перплексия
3. Метрика уникальности ядер

### 3.6 Недостатки реализации

Недостатком реализации является невозможность промышленного применения, поскольку реализация изначально была ориентированна на удобное и

гибкое проведение экспериментов. Невозможность промышленного применения означает, что существуют участки кода программы, которые нуждаются в оптимизации, перепроектировании или рефакторинге.

### 3.7 Процесс построения тематической модели

Процесс построения модели можно наглядно представить следующим образом (Рис. 9): первым этапом происходит сбор коллекции, чаще всего из открытых источников, далее над коллекцией производятся преобразования описанные в разделе 1.7, далее по подготовленной коллекции строятся тематические модели, вычисляются метрики качества, далее результаты анализируются для подготовки новых экспериментов.

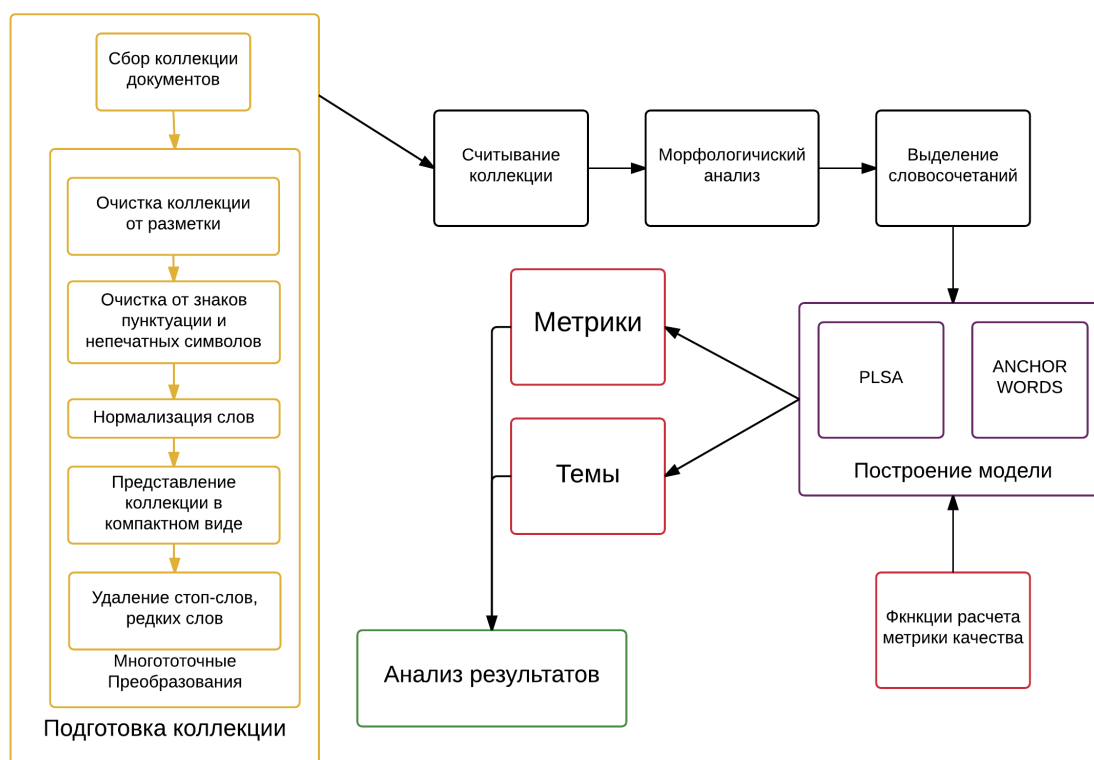


Рис. 9: Схема процесса построения тематической модели

## Пример запуска эксперимента для построения модели

В Листинге 7 приведён пример построения комбинации моделей PLSA и Anchor Words, с использованием биграмм. Все модули приложения реализованы в виде python пакетов.

```
1 from tmtk.topic_models import anchor, plsa
2
3 from tmtk.metrics.metrics import *
4 from tmtk.collection.collection import FullTextCollection
5
6 from tmtk.collection.transformer_api import TransformerChainApply
7 from tmtk.collection.transformer import *
8
9 collection = FullTextCollection(
10     path='./tmtk/corpa/ru_bank_wid_small.zip', lang='ru'
11 ).fill()
12
13 transformers = TransformerChainApply(
14     transformers=[
15         BigramExtractorDocumentsTransform(do_apply=False)
16     ]
17 )
18 collection = transformers.apply(collection)
19
20 F, anc = anchor.anchor_model(
21     collection, wrd_count=collection.num_wrd, bi=True
22     metrics=[preplexity, coherence, uniq_top_of_topics])
23
24 anchor.print_topics(F, collection.id_to_words, anc, 'an+bi.txt')
25
26 F, T = plsa.plsa_model(
27     collection, wrd_count=collection.num_wrd,
28     metrics=[preplexity, coherence, uniq_top_of_topics],
29     num_iter=5, verbose=True, F=F)
30
31 plsa.print_topics(F, collection.id_to_words, 'an+pl+bi.txt')
```

Листинг 7: Пример запуска эксперимента для построения модели

## 4 Учет лингвистических знаний в тематических моделях

### 4.1 Постановка задачи

Требуется разработать модификации стандартных моделей, с более высокой уникальностью ядер, засчёт интеграции дополнительных параметров: учета частей речи и словосочетаний.

Реализовать данную модификацию и провести эксперименты и выполнить оценку качества моделей.

Цель данного раздела — показать, что учёт лингвистических знаний может улучшить качество вероятностных тематических моделей, основанных только на частотах употребления слов.

Эксперименты, приведённые в данном разделе, проведены на коллекции текстов банковской тематики на русском языке, взятых из электронных журналов ([goo.gl/L97CZ8](http://goo.gl/L97CZ8)). Предварительно слова в текстах были нормализованы, приведены к нижнему регистру. Было проведено удаление стоп-слов, тематически не окрашенных слов (по морфологическим характеристикам), знаков пунктуации.

Коллекция размером в 2500 документов была случайным образом разделена на 2000 документов для обучения и 500 для контроля.

Для вычисления *перплексии* на тестовой коллекции документ случайным образом разделялся на две части (случайным образом выбирались слова из документа), после чего первая часть использовалась для оценки распределения тем в документе, а вторая для оценки перплексии.

Для представление результатов экспериментов использованы следующие обозначения:

1. **an** — алгоритм Anchor Words.
2. **pl** — алгоритм PLSA.
3. **bi** — использование словосочетаний, биграмм.
4. **mr** — использование морфологических ограничений.
5. **1.23** — нестабильный результат.
6. **1.23** — лучший результат.

Результат для алгоритма выбирался фиксацией метрик на лучшей итерации, лучшей по перплексии на контроле (иногда почти самой лучшей в пределах двух итераций от лучшего значения).

Реализация экспериментов и алгоритмов доступна по адресу [goo.gl/f8ZdcW](http://goo.gl/f8ZdcW).

## 4.2 Комбинация Anchor Words и PLSA

### Anchor Words как начальное приближение для PLSA

Алгоритмы Anchor Words и PLSA могут устранить недостатки друг друга: модель Anchor Words в отличие от PLSA не пользуется случайным приближением матриц  $\Phi$  и  $\Theta$ , а руководствуется достаточно слабым предположением  $p$ -разделимости, модель PLSA оптимизирует перплексию, которая является недостатком Anchor Words.

Возникает предположение, что модель Anchor Words будет хорошим начальным приближением для PLSA. Проведённые эксперименты подтвердили данное предположение одновременным улучшением сразу нескольких метрик

качества. Данное предположение уже обсуждалось в литературе, и не является идеей автора, но дальнейшие эксперименты будут проведены именно с использованием комбинации моделей. Алгоритм обучения модели приведён в Листинге. 6. В таблице [2, 3, 5] приведены примеры тем, полученных в результате проведения эксперимента.

---

### Алгоритм 6 Anchor Words + PLSA

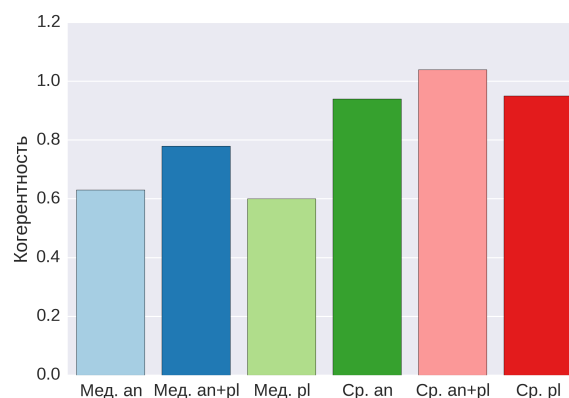
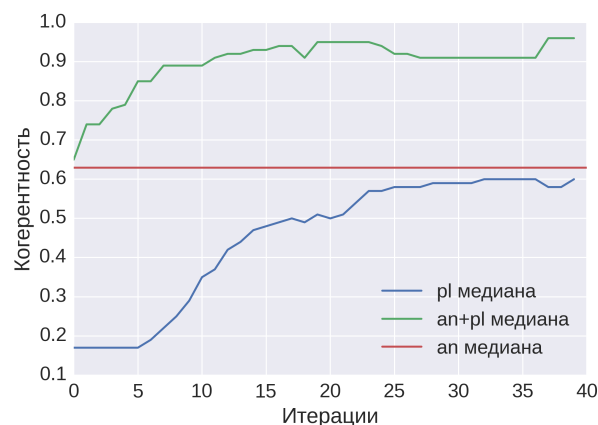
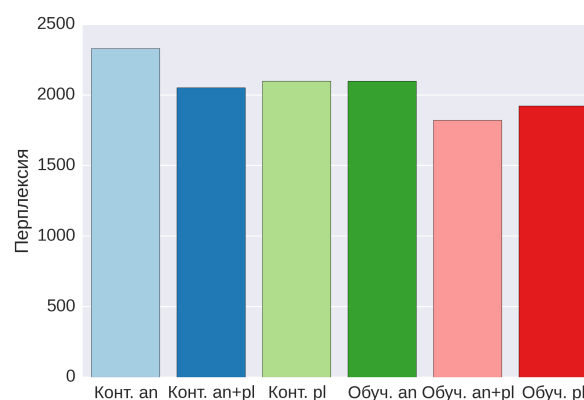
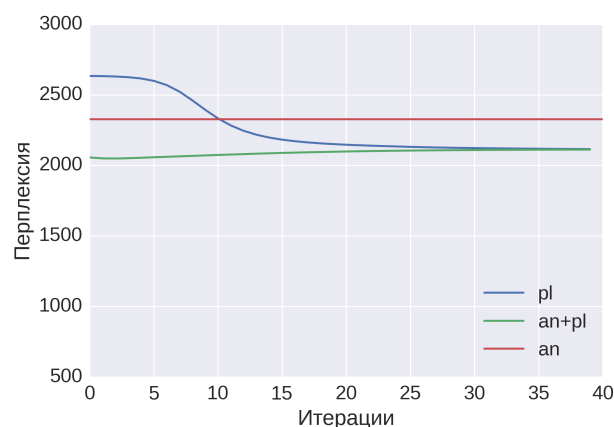
---

**Вход:** коллекция  $D$ , число тем  $|T|$ ;

**Выход:** матрица  $\Phi, \Theta$ ;

- 1:  $F = \text{Anchor\_Words}(D, T)$
  - 2:  $F, \Theta = \text{PLSA}(D, T, F)$
  - 3: **Вернуть**  $F, \Theta$
- 

## Эксперименты



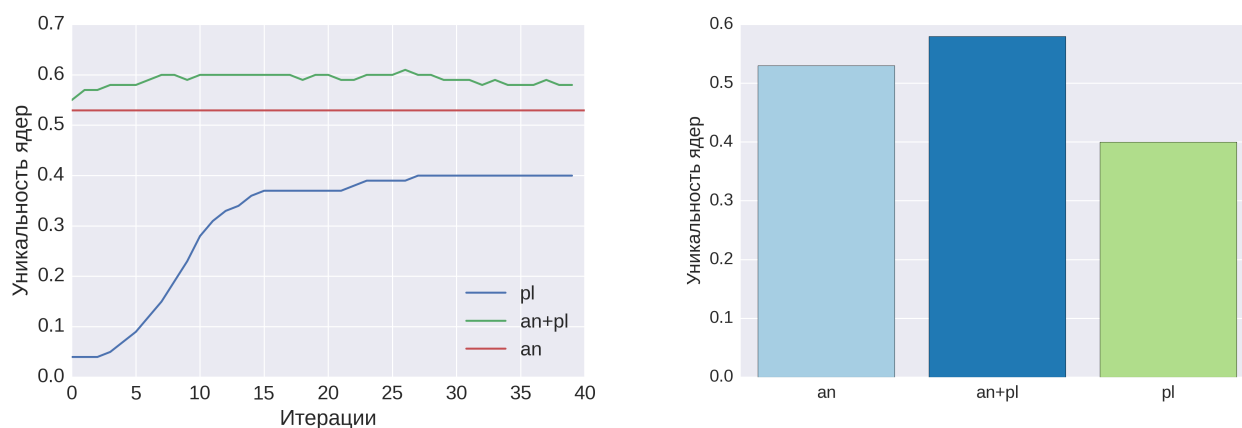


Рис. 10: Результаты экспериментов, комбинация алгоритмов Anchor Words, PLSA, Anchor Words + PLSA.

### 4.3 Поиск кандидатов в Anchor Words

#### Лингвистические требования к тематически окрашенным терминам

В статье [Aroga2012b] на термины, которые могли стать якорными накладывались частотные ограничения для устранения шумов.

Но термины помимо своих частотных характеристик обладают также и лингвистическими характеристиками – морфологическими, грамматическими и другими.

Было выдвинуто предположение, что якорными словами могли стать те термины, которые прошли частотный отбор и являются существительными (Листинг. 7).

Эксперименты показали что при добавлении морфологических ограничений перплексия уменьшается, средняя интерпретируемость возрастает, но медиана интерпретируемости и уникальность ядер ведет себя нестабильно.



---

**Алгоритм 7** Комбинаторный алгоритм FastAnchorWords с фильтрацией существительных.

---

**Вход:** Точки  $V = v_1, \dots, v_n$ , количество векторов для выпуклой оболочки –  $K$ ;

**Выход:**  $\{v'_1, \dots, v'_k\}$  – точки образующие наиболее выпуклую оболочку;

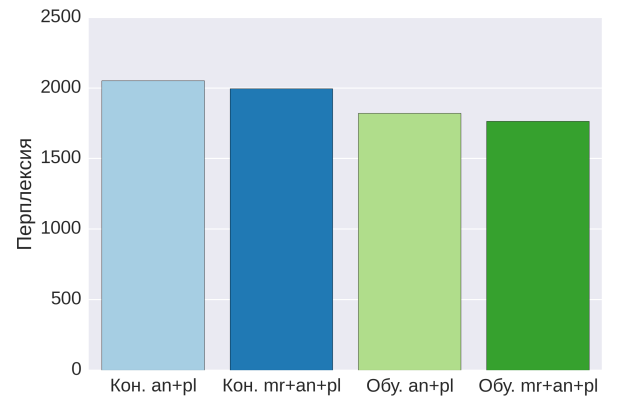
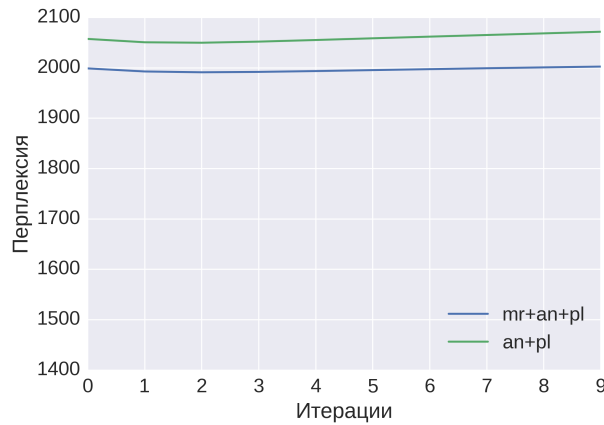
- 1: Отфильтровать только те  $v_i$ , которые являются существительными.
- 2: Спроецировать  $v_i$  на случайно выбранное пространство  $V$ ,  $\dim V = 4 \log V / \epsilon^2$
- 3:  $S = \{s_0\}$ ,  $s_0$  – наиболее удалённая точка от начала координат.
- 4: **для всех**  $i = 1$  to  $K$ :
- 5:     Обозначим  $s_i$  точку из  $V$  наиболее удалённую от  $\text{span}(S)$ .
- 6:      $S = S \cup \{s_i\}$
- 7: **для всех**  $i = 1$  to  $K$ :
- 8:     Обозначим  $s'_i$  точку из  $V$  наиболее удалённую от  $\text{span}(S \setminus \{s_i\})$ .
- 9:     Заменить  $s_i$  на  $s'_i$

**Вернуть**  $S$

---

В таблице [4] приведены примеры тем, полученных в результате проведения эксперимента.

## Эксперименты



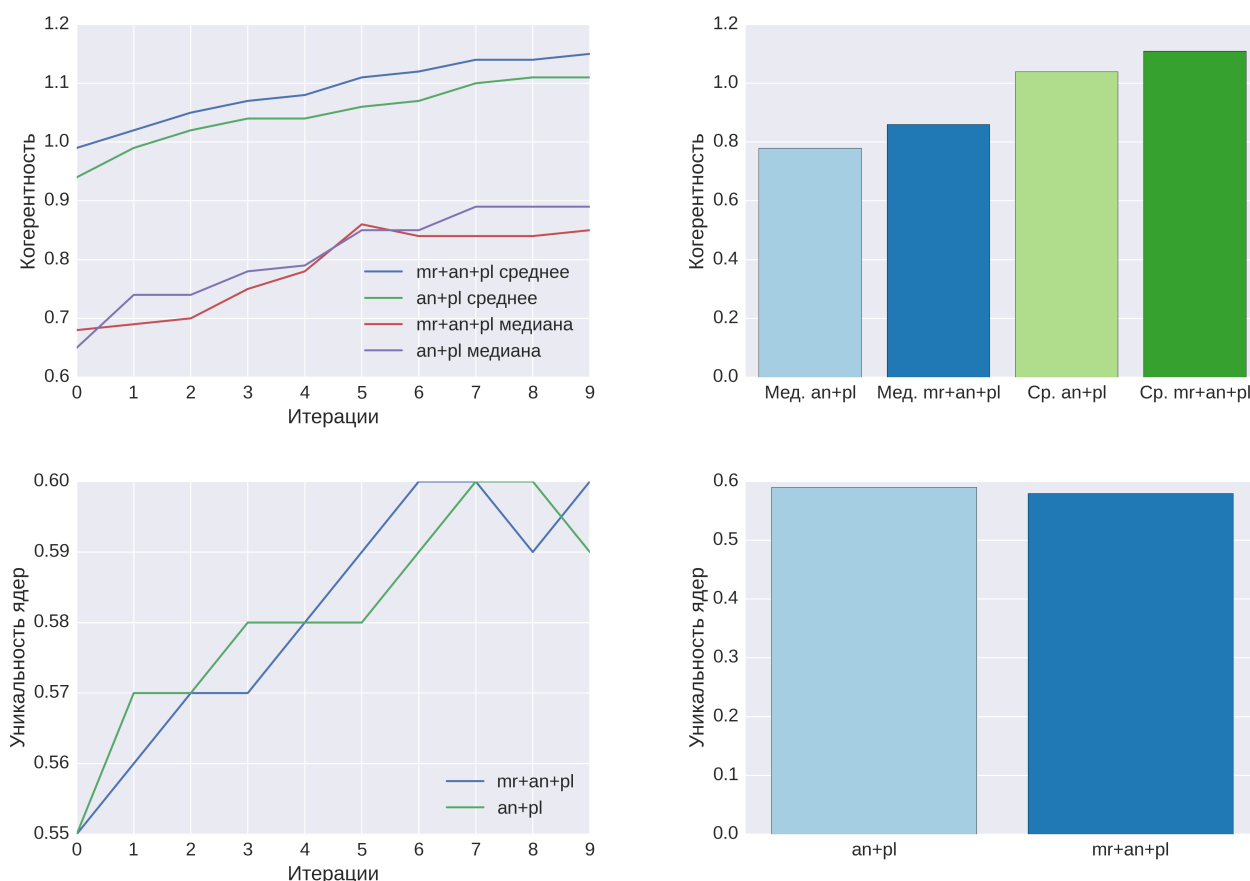


Рис. 11: Результаты экспериментов, Anchor Words + PLSA, Anchor Words + PLSA + Morph.

## 4.4 Учёт словосочетаний в Anchor Words

Модель мешка слов совсем не учитывает порядок слов в документах, а на самом деле многие термины, употреблённые в тексте, употреблены именно в словосочетаниях и имеют совершенно другую тематическую окраску.

Добавление словосочетаний как уникальных элементов словаря существенно ухудшает перплексию, за счёт увеличения размера словаря, но увеличивает интерпретируемость тем. Возникает вопрос — можно ли учесть словосочетания в алгоритме Anchor Words, не добавляя их как уникальные элементы словаря?

## Выделение словосочетаний (биграмм)

Для выделения словосочетаний был использован метод предложенный в [Dobrov2001].

Авторы предлагают следующий алгоритм — Если *автор текста* использует некоторый термин, как отдельную единицу изложения, опирается на этот термин в своём изложении, то именно в этом тексте слова термина рядом будут встречаться чаще, чем в разбивку. Далее предполагается, что если пара слов встречается как непосредственные соседи более чем в половине случаев их появления в одном и том же текстовом окне, то это свидетельствует о том, что эта пара слов в совокупности служит автору опорной точкой, то есть представляет собой термин или фрагмент термина.

Псевдокод алгоритма представлен в Листинге 8.

---

### Алгоритм 8 Anchor Words + PLSA

---

**Вход:** коллекция  $D$ , ширина окна  $k$ ;

**Выход:** 1000 самых частотных биграмм;

```
1:  $Bigramms = Set()$ 
2: для всех  $d \in D$ :
3:   для всех  $w_i, w_j$ :
4:      $n_{neib}$  = количество окон в которых,
5:     пара слов  $w_i, w_j$  встречались как соседи.
6:      $n$  = количество окон в которых,
7:     пара слов  $w_i, w_j$  встречалась одновременно.
8:     Если  $n_{neib} \geq 0.5 * n$  тогда:
9:        $Bigramms += (w_i, w_j)$ 
10: Вернуть 1000 самых часто встречаемых словосочетаний из  $Bigramms$ .
```

---

Для дальнейшего использования в тематических моделях предполагается использовать 1000 самых частотных биграмм.

## Представление словосочетаний в Anchor Words

Существует предположение, что биграммы точнее определяют тему, чем униграммы, поэтому возникла гипотеза, что биграммы будут хорошими якорными словами. Но добавление биграмм как уникальных элементов словаря ухудшает перплексию, поэтому был предложен следующий метод.

Обозначим  $B = \{(w_1, w_2)_1, \dots, (w_1, w_2)_n\}$  – множество биграмм,  $W = \{w_1, \dots, w_n\}$  – вектора соответствующие терминам (в алгоритме 4 ковариационная матрица слов), тогда представим каждую биграмму, как сумму векторов слов образующих данную биграмму –  $W(w_1) + W(w_2)$ . Произведём поиск якорных слов среди

$$Q = \{q_1, \dots, q_n, W(b_{11}) + W(b_{12}), \dots, W(b_{n1}) + W(b_{n1})\},$$

после чего продолжим алгоритм 4 без изменений.

---

### Алгоритм 9 Высокоуровневый алгоритм Anchor Words с учетом биграмм.

---

**Вход:** коллекция  $D$ , число тем  $|T|$

**Выход:** матрица  $\Phi$ ;

- 1:  $Q = \text{Word Co-occurences}(D)$
  - 2:  $\hat{Q} = \text{Rows normalized } Q$
  - 3:  $\hat{Q} = \{\hat{Q}_0, \dots, \hat{Q}_n, \hat{Q}[BI_1[1]] + \hat{Q}[BI_1[2]], \hat{Q}[BI_i[1]] + \hat{Q}[BI_i[2]]\}$
  - 4:  $S = \text{FindAnchorWords}(\hat{Q}, |T|)$
  - 5:  $\hat{Q} = \{\hat{Q}_0, \dots, \hat{Q}_n\}$
  - 6:  $\Phi = \text{RecoverWordTopic}(\hat{Q}, S)$
  - 7: **Вернуть**  $\Phi$
- 

Проведённые эксперименты подтвердили данное предположение одновременным улучшением сразу нескольких метрик качества. В таблице [5] приведены примеры тем, полученных в результате проведения эксперимента.

## Эксперименты

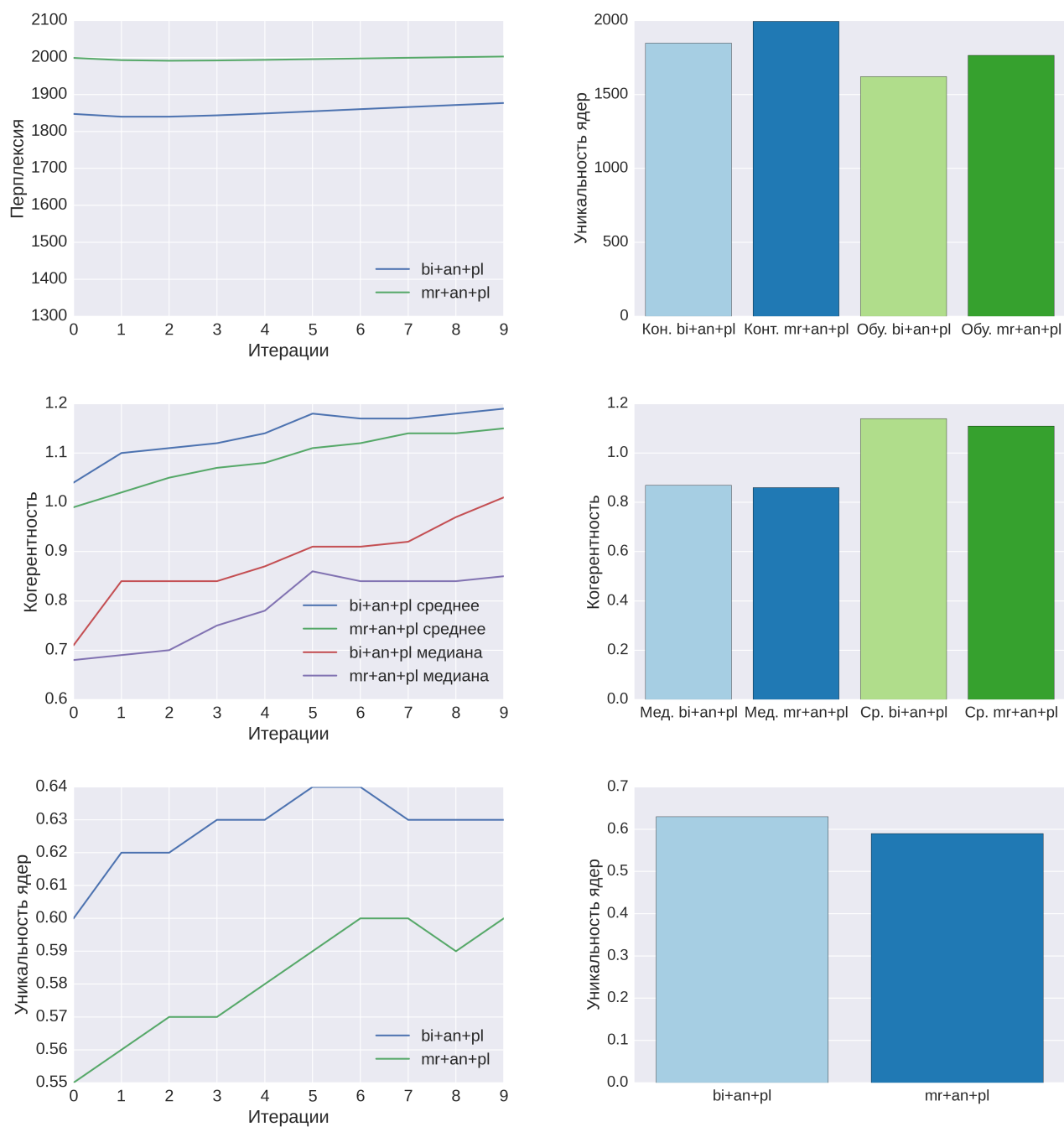


Рис. 12: Результаты экспериментов, Anchor Words + PLSA + Morph, Anchor Words + PLSA + Bigramm.

Хочется отметить, что при проведении экспериментов большинство якорных слов соответствовало биграммам.

## 4.5 Сравнение результатов

В данном разделе результаты экспериментов объединены в одну таблицу. Учёт лингвистических предположений дал заметное улучшение метрик качества, лучший результат показала комбинация **Anchor Words + PLSA + BIGRAMMS**, причём минимум перплексии достигался уже на первых итерациях PLSA, в следствии чего комбинация работала значительно быстрее, чем оригинальный PLSA.

Использование словосочетаний помогает улучшить качество тематических моделей, но все равно не учитывает большую часть информации. Безусловно будущее тематического моделирования – отказ от гипотезы мешка слов.

Также, вполне разумным и эффективным является наложение лингвистических ограничений на якорные слова.

Таблица 1: Сравнительная таблица метрик качества различных алгоритмов

Алгоритм	Перплексия		Когерентность		Уник. ядер
	обуч.	конт.	сред.	меди.	
PL	1923	2116	0.95	0.60	0.40
AN	2099	2330	0.94	0.63	0.53
AN+PL	1822	2052	1.04	0.78	0.58
MR+AN+PL	1765	1995	1.11	0.86	0.59
BI+AN+PL	<b>1621</b>	<b>1848</b>	<b>1.14</b>	<b>0.87</b>	<b>0.63</b>

- PL — вероятностный латентный семантический анализ.
- AN — алгоритм якорных слов.
- MR — морфологические ограничения на якорные слова.
- BI — учёт словосочетаний.

# Заключение

Для выполнения данной работы были изучены и реализованы стандартные модели *тематического моделирования* — PLSA и Anchor Words.

Предложены подходы к построению тематических моделей, который позволил улучшить интерпретируемость и уникальность статистических тем:

1. Внедрение морфологических ограничения в алгоритм Anchor Words
2. Учёт словосочетаний в алгоритме Anchor Words
3. Введена метрика уникальности ядер

Проведены эксперименты, показавшие улучшение метрик качества тематических моделей. Полученные результаты показывают, что внедрение словосочетаний дало наибольший прирост из рассматриваемых подходов, сразу по всем рассматриваемым метрикам качества.

В дальнейшем предполагается проверить работоспособность приведённых подходов на коллекциях текстов английского языка.

# Список литературы

- [AnkurMoitraSlides2012] Ankur Moitra, joint with Sanjeev Arora and Rong Ge, 2012 *Learning Topic Models – Going Beyond SVD*, slides
- [Arora2012] Sanjeev Arora, Rong Ge, Ankur Moitra, 2012 *Learning Topic Models Going beyond SVD*, Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium
- [Arora2012b] Sanjeev Arora, Rong Ge, David Sontag, Yoni Halpern, Yichen Wu, David Mimno, Michael Zhu, Ankur Moitra, 2012 *A Practical Algorithm for Topic Modeling with Provable Guarantees*, <http://arxiv.org/abs/1212.4777>
- [ARTM] Воронцов К. В., Потапенко А. А. *Регуляризация вероятностных тематических моделей для повышения интерпретируемости и определения числа тем.*
- [Blei2003] Blei D. M., Ng A. Y., Jordan M. I. *Latent Dirichlet allocation*, Journal of Machine Learning Research 2003.
- [Dobrov2001] Б.В.Добров, Н.В.Лукашевич, С.В.Сыромятников *Формирование базы терминологических словосочетаний по текстам предметной области*
- [Newman2009] David Newman, Jey Han Lau, Karl Grieser, Timothy Baldwin *Automatic Evaluation of Topic Coherence*, 2009.
- [Nokel2014] М. А. Нокель *Тематические модели: учет сходства между униграммами и биграммami*, Russian Conference on Digital Libraries 2014.



- [Nyzhybytskyu2014] Нижибицкий Е.А. *Относительная перплексия как мера качества тематических моделей*
- [Hofmann1999] Thomas Hofmann, *Probabilistic Latent Semantic Analysis*, 1999.
- [Kreinovich2005] Kreinovich V., Kearfott R.B. 2005, *Beyond convex? Global optimization is feasible only for convex objective functions*
- [Voron2013] Воронцов К. В., курс лекций *Вероятностные тематические модели*, 2013.
- [Voss2014] Catalin Voss, *Building Topic Models Based on Anchor Words*, 2014
- [VoronSlides2015] Воронцов К. В., слайды для курса лекций *Вероятностные тематические модели*, 2015.

Таблица 2: Пример тем, алгоритм PLSA

Topic 0	денежный инфляция производство страна экономика экономический рост год цена
Topic 1	объект фактор рубль сумма клиент оценка стоимость время факторинг
Topic 2	налоговый рф орган налог ст налогоплательщик законодательство закон право
Topic 3	стандарт международный финансовый страна год система национальный вопрос организация
Topic 4	страна китаи государственный США система китайский доллар международный экономика
Topic 5	бюджетный средство финансовый программа бюджет развитие федеральный проект государственный
Topic 6	вклад страхование банковский система банка банк организация услуга депозит
Topic 7	надзор банковский орган принцип надзорный банка капитал базель риска
Topic 8	год бизнес банка банк розничный малое работать клиент развитие
Topic 9	модель метод инновационный оценка анализ показатель являться уровень исследование
Topic 10	реклама рекламный год говорить банка очень время продукт статья
Topic 11	капитал средство экономика актив экономический собственный ресурс форма увеличение
Topic 12	банка банк бренд банкир год банковский очень клиент российский
Topic 13	самый год отель самолет офшорный место километр мир трасса
Topic 14	кредит банка рынок банк кредитование год потребительский агентство считать
Topic 15	ассоциация россия КБ банк банковский москва мурычев совет ОАО
Topic 16	государственный система экономический наука защитить экономика библиогра назый экон
Topic 17	внутренний контроль организация управление система аудит подразделение служба внешний
Topic 18	монета денежный обращение россия рубль год деньга реформа золото
Topic 19	сотрудник работа персонал система обучение специалист программа управление документ
Topic 20	деньга стоимость цена товар оценка компания рыночный функция денежный

Таблица 3: Пример тем, алгоритм Anchor Words

москва	москва банка банк ооо россия лицензия номер регистрация
налоговый	налоговый налог рф ст налогоплательщик порядок уплата сумма
история	кредитный история бюро информация организация замщик закон банка
нормативный	банка россия нормативный закон банковский надзор деятельность федеральный
председатель	председатель правление банк оао кб зао акб главный
основание	организация кредитный россия основание государственный банка управление регион
важный	важный потребитель доверие завоевание упрочение связь время являться
валюта	валюта рубль курс иностранный национальный использовать банк капитал
управлять	компания управлять адрес год решение фактический требование телефон
рабочий	рабочий время работа неделя норма группа труд работник
ставка	ставка процентный рефинансирование уровень депозит цб годовой снижение
отчтность	отчтность финансовый мсфо организация бухгалтерский аудит аудитор вопрос
страхование	страхование страховой страхов страховщик обязательный премия участник компания
опыт	опыт международный страна рынок работа тема поддержка иностранный
акция	акция рынок размещение акционер инвестор капитал фондовый биржа
сила	россия банка сила изменение указание официальный год положение
платж	платж система платжный счет расчт карта средство услуга
ассоциация	ассоциация банковский банк россия российский президент банка арба
защита	защита система информация решение отношение безопасность право дать
юридический	лицо юридический физический счет реестр деятельность сведение право
фонд	фонд пенсионный негосударственный пенсия год накопление средство тысяча

Таблица 4: Пример тем, алгоритм  
Anchor Words + PLSA

москва	москва ооо кб оао банк банка адрес зао номер
налоговый	налоговый налог ст рф налогоплательщик налогообложение уплата кодекс порядок
история	кредитный история информация замщик бюро закон организация риск запрос
нормативный	банка закон надзор федеральный деятельность порядок нормативный банковский акт
председатель	председатель правление оао александр владимир главный банк зао кб
основание	организация кредитный основание государственный регион внести управление приказ регистрация
важный	важный потребитель доверие время должный канал необходимый позволять особенно
валюта	рубль валюта курс иностранный национальный использовать укрепление капитал евро
управлять	компания управлять управление решение год инвестиционный фактический ук требование
рабочий	рабочий работник труд время работа трудовой женщина выходной продолжительность
ставка	ставка процентный уровень рефинансирование снижение инфляция ликвидность депозит высокий
отчетность	отчетность мсфо финансовый оценка бухгалтерский информация составление организация должный
страхование	страхование страховой страхов страховщик обязательный премия компания выплата тариф
опыт	страна международный рынок опыт работа развивающийся поддержка иностранный германия
акция	акция рынок размещение акционер инвестор фондовый биржа доля втб
сила	россия сила изменение банка положение указание официальный год внесение
платж	платж платжный расчт система счт карта средство услуга использование
ассоциация	ассоциация банковский президент арба банк российский развитие региональный россия
защита	защита система отношение безопасность информация решение доступ дать угроза
юридический	лицо юридический физический счт право сведение реестр форма деятельность
фонд	фонд пенсионный пенсия негосударственный накопление средство обеспечение инвестиционный год

Таблица 5: Пример тем, алгоритм  
Anchor Words + PLSA + Morph

москва	москва банк банка россия оао ооо адрес кб зао
история	кредитный история информация бюро замщик закон организация отчт запрос
затрата	затрата метод управленческий необходимый продукция использовать себестоимость дать производственный
председатель	председатель правление банк оао александр владимир кб зао московский
основание	организация кредитный основание россия государственный внести следующий регистрация приказ
осуществление	банковский банк операция осуществление кредитный россия банка лицензия организация
миллиард	миллиард рубль миллион составить увеличиться сумма актив средство доля
связь	связь важный потребитель мера россия доверие учитывать общественный внешний
страхование	страхование страховой страхов страховщик обязательный премия выплата вид тариф
валюта	валюта валютный рубль курс денежный иностранный центральный национальный резерв
опыт	международный рынок страна опыт банк развивающийся тема германия иностранный
отчетность	отчетность бухгалтерский финансовый мсфо учт аудитор аудиторский учтный организация
фонд	фонд пенсионный пенсия управлять накопление негосударственный инвестиционный средство инвестирование
ставка	ставка процентный рефинансирование ликвидность уровень рынок снижение депозит инфляция
сила	россия сила изменение банка положение указание год официальный порядок
ассоциация	ассоциация банковский банк россия арба российский президент сообщество региональный
филиал	филиал банка банк кредитный офис сбербанк миллион отделение открытие
защита	защита система безопасность дать обеспечение информация проблема решение доступ
акция	акция размещение акционер инвестор фондовый предложение доля пакет приобретение
рабочий	рабочий работник труд время трудовой женщина место часы мужчина
совет	совет деятельность россия принять минфин заседание комитет вопрос ответственный

Таблица 6: Пример тем, алгоритм  
Anchor Words + PLSA + Bigramm

пенсионный_резерв	пенсионный пенсия негосударственный накопление взнос обеспечение размер пенсионер
обособленный_подразделение	подразделение сотрудник работа руководитель специалист персонал внедрение обучение
платжный_карта	карта платжный операция использование услуга банкомат сеть количество пластиковый
важный_роль	фактор национальный являться степень важный инновационный наиболее условие потенциал
валютный	валютный валюта курс операция центральный ограничение экономика резерв регулирование
рубль	рубль сумма валюта составить месяц тысяча выпуск годовой январь
наиболее_важный	являться фактор роль уровень необходимый условие процесс результат степень
товар	товар услуга поставщик заказ поставка торговый запас вариант торговля
уплата_налог	налог сумма доход налогообложение прибыль уплата период бюджет минфин
хозяйствовать_субъект	экономический экономика субъект наука современный библиогра экон
важный_фактор	процесс теория уровень исследование показатель роль значение модель подход
договор_купипродажа	договор условие имущество обязательство случай ст обеспечение соответствие требование
вклад	вклад банк вкладчик день депозит годовой банка физический срок
директор	директор генеральный группа руководитель должность начальник деловой заместитель
банковский	банковский банк надзор сектор развитие услуга цб деятельность повышение
денежный	денежный перевод обращение средство масса проведение мигрант выпуск поток
очень_важный	проблема хороший говорить достаточно слово сделать разный большинство взгляд
малое_среднее	бизнес развитие малое регион среднее поддержка региональный средний предприниматель
образовательный_учреждение	развитие учреждение образование высокий условие направление цель образовательный
ставка_рефинансирование	ставка процентный рефинансирование снижение год уровень рост изменение ликвидность
ипотечный_кредитование	кредитование ипотечный ипотека жиль жилищный банк розничный ресурс проблема